

This model and documentation are provided for non-commercial use only under the CC BY-NC-SA 4.0 license.
No commercial use, profiling, or exploitation of IP logs or trade secrets is permitted.
See full “License and Conditions of Use” notice for details.
In case of conflict or ambiguity, the most restrictive interpretation in favor of non-commercial use, user privacy, and protection of trade secrets shall prevail.
Author: Carlos Daniel Borrego Romero

Annex B — Reference Test Vector and Validation Pseudocode for QHD-1.0

Purpose

This annex provides a reference test vector and implementation-oriented validation pseudocode for the Earth base case under QHD-1.0. Its purpose is to support direct replication of the validation workflow documented in the Earth proof-of-concept record, including entropy normalization, continuous descriptor construction, canonical JSON serialization, and SHA3-256 hash generation [cite:55][cite:56][cite:57].

The annex is designed to reduce implementation ambiguity. In practice, deterministic identity frameworks fail less often because of equations than because of differences in numeric formatting, key ordering, unit labeling, or time handling, so those details are made explicit here [cite:94][cite:96].

Reference assumptions

The following assumptions define the Earth reference test vector under QHD-1.0.

Field	Value
Object name	Earth
Object class	planet
Reference frame	SSB
UTC epoch	2024-01-01T00:00:00Z
TAI-UTC offset	37 s [cite:46][cite:49][cite:52]
Serialized time field	1704067237 [cite:57]
Mass field	5.9722e24 kg [cite:18]
Informational partition	core = 0.31, mantle = 0.68, crust = 0.01 [cite:43]
K	1.0
γ	1.0

This model and documentation are provided for non-commercial use only under the CC BY-NC-SA 4.0 license.

No commercial use, profiling, or exploitation of IP logs or trade secrets is permitted.

See full “License and Conditions of Use” notice for details.

In case of conflict or ambiguity, the most restrictive interpretation in favor of non-commercial use, user privacy, and protection of trade secrets shall prevail.

Author: Carlos Daniel Borrego Romero

S_{ref}	0.5
$C(t)$	1
Serialization version	QHD-1.0
Hash algorithm	SHA3-256 [cite:7][cite:13]

These assumptions should be treated as frozen for validation purposes. Any change to the informational partition, parameter defaults, field names, precision policy, or time conversion rule creates a different test condition and should therefore be versioned explicitly rather than silently substituted [cite:56][cite:96].

Reference numerical outputs

Using the declared Earth assumptions, the following intermediate and final outputs must be reproduced by a conforming implementation [cite:57].

Quantity	Expected value
S_{norm}	0.6111063202
$\log_{10}(E_0)$	41.7297757490
$\text{Re}(H_{\text{cont}})$	41.7308868122
$\text{Im}(H_{\text{cont}})$	0.6111063202
Canonical H_{cont} .Re string	4.17308868e+01
Canonical H_{cont} .Im string	6.11106320e-01
Canonical M_{kg} string	5.97220000e+24
SHA3-256 digest	4be65a8add8b7b74b184afe2b0b3ef7d068276e6e8ea1f399a75adb630e23510

The expected values above provide a direct pass/fail benchmark for software validation. If an implementation reproduces the final hash but not the intermediate quantities, the implementation should still be audited because the agreement may be accidental or may conceal undocumented formatting behavior [cite:96].

This model and documentation are provided for non-commercial use only under the CC BY-NC-SA 4.0 license.

No commercial use, profiling, or exploitation of IP logs or trade secrets is permitted.

See full “License and Conditions of Use” notice for details.

In case of conflict or ambiguity, the most restrictive interpretation in favor of non-commercial use, user privacy, and protection of trade secrets shall prevail.

Author: Carlos Daniel Borrego Romero

Reference test vector

The following canonical JSON object is the normative Earth reference test vector for QHD-1.0 under the declared validation policy [cite:55][cite:56].

```
{ "C": 1, "H_cont": { "Im": "6.11106320e-01", "Re": "4.17308868e+01" }, "M_kg": "5.97220000e+24", "T_TAI_unix_s": 1704067237, "epoch_utc": "2024-01-01T00:00:00Z", "object_class": "planet", "object_name": "Earth", "reference_frame": "SSB", "serialization_version": "QHD-1.0" }
```

A conforming implementation must serialize the object exactly as above before applying SHA3-256.

Any deviation in numeric formatting, whitespace policy, key ordering, field spelling, or time representation will generally produce a different digest and should therefore be treated as a failed conformance case unless the protocol version explicitly permits the difference [cite:94][cite:96].

Validation pseudocode

The following pseudocode is language-agnostic and is intended to make the validation logic explicit for independent implementations.

INPUTS:

```
object_name = "Earth"
object_class = "planet"
reference_frame = "SSB"
epoch_utc = "2024-01-01T00:00:00Z"
TAI_minus_UTC = 37
unix_utc_seconds = 1704067200
M_kg = 5.9722e24
p = [0.31, 0.68, 0.01]
K = 1.0
gamma = 1.0
S_ref = 0.5
C = 1
serialization_version = "QHD-1.0"
c = 299792458.0
```

STEP 1: Validate the probability state

```
assert sum(p) == 1 within tolerance
```

```
assert every p_i >= 0
```

This model and documentation are provided for non-commercial use only under the CC BY-NC-SA 4.0 license.

No commercial use, profiling, or exploitation of IP logs or trade secrets is permitted.

See full “License and Conditions of Use” notice for details.

In case of conflict or ambiguity, the most restrictive interpretation in favor of non-commercial use, user privacy, and protection of trade secrets shall prevail.

Author: Carlos Daniel Borrego Romero

```
N = length(p)
```

STEP 2: Compute entropy

```
S = -sum over i of p_i * ln(p_i)
```

```
S_max = ln(N)
```

```
S_norm = S / S_max
```

STEP 3: Compute energy anchor

```
E0 = M_kg * c^2
```

```
log10_E0 = log10(E0)
```

STEP 4: Compute continuous descriptor

```
Re_H_cont = log10_E0 + (K / 100) * (S_norm - S_ref)
```

```
Im_H_cont = gamma * S_norm
```

STEP 5: Compute serialized time

```
T_TAI_unix_s = unix_utc_seconds + TAI_minus_UTC
```

STEP 6: Format scalar strings

```
Re_string = scientific_notation(Re_H_cont, 8 digits after decimal)
```

```
Im_string = scientific_notation(Im_H_cont, 8 digits after decimal)
```

```
M_string = scientific_notation(M_kg, 8 digits after decimal)
```

STEP 7: Build canonical object

```
canonical = {  
  "C": C,  
  "H_cont": {  
    "Im": Im_string,  
    "Re": Re_string  
  },  
  "M_kg": M_string,  
  "T_TAI_unix_s": T_TAI_unix_s,  
  "epoch_utc": epoch_utc,  
  "object_class": object_class,  
  "object_name": object_name,  
  "reference_frame": reference_frame,  
  "serialization_version": serialization_version  
}
```

STEP 8: Canonical JSON serialization

This model and documentation are provided for non-commercial use only under the CC BY-NC-SA 4.0 license.

No commercial use, profiling, or exploitation of IP logs or trade secrets is permitted.

See full “License and Conditions of Use” notice for details.

In case of conflict or ambiguity, the most restrictive interpretation in favor of non-commercial use, user privacy, and protection of trade secrets shall prevail.

Author: Carlos Daniel Borrego Romero

```
json_string = serialize_JSON(  
    object = canonical,  
    sort_keys = true,  
    separators = [",", ":"],  
    encoding = UTF-8,  
    no_extra_whitespace = true  
)
```

STEP 9: Hash generation

```
digest = SHA3_256(json_string as UTF-8 bytes)  
digest_hex = lowercase_hex(digest)
```

STEP 10: Validation checks

```
assert round_or_format(S_norm) matches 0.6111063202 under policy  
assert round_or_format(log10_E0) matches 41.7297757490 under policy  
assert Re_string == "4.17308868e+01"  
assert Im_string == "6.11106320e-01"  
assert M_string == "5.97220000e+24"  
assert T_TAI_unix_s == 1704067237  
assert json_string matches the reference test vector exactly  
assert digest_hex == "4be65a8add8b7b74b184afe2b0b3ef7d068276e6e8ea1f399a75adb630e23510"
```

OUTPUTS:

```
S_norm  
log10_E0  
Re_H_cont  
Im_H_cont  
json_string  
digest_hex
```

Conformance logic

A software implementation should be considered conformant for this Earth test case only if it reproduces both the canonical JSON and the final SHA3-256 digest exactly [cite:55][cite:57][cite:96]. Matching only the floating-point values without matching the canonical serialization is insufficient for protocol conformance, because the QHD identity layer depends on the exact serialized representation rather than on abstract numerical equivalence alone [cite:94][cite:96].

This model and documentation are provided for non-commercial use only under the CC BY-NC-SA 4.0 license.
No commercial use, profiling, or exploitation of IP logs or trade secrets is permitted.
See full “License and Conditions of Use” notice for details.
In case of conflict or ambiguity, the most restrictive interpretation in favor of non-commercial use, user privacy, and protection of trade secrets shall prevail.
Author: Carlos Daniel Borrego Romero

A useful validation sequence is the following:

1. Verify the entropy state and intermediate scalar outputs.
2. Verify the formatted scalar strings.
3. Verify the canonical JSON byte-for-byte.
4. Verify the SHA3-256 digest.
5. Repeat the full run in a second implementation or execution environment [cite:96].

This layered procedure helps distinguish mathematical errors from serialization errors and makes debugging easier during proof-of-concept development [cite:96].

Failure modes

The most common causes of failed reproduction in a deterministic identity pipeline are procedural rather than theoretical [cite:94][cite:96]. Typical failure modes include:

- Using different logarithm bases or inconsistent entropy normalization.
- Allowing floating-point fields to serialize as native numbers instead of fixed scientific-notation strings.
- Using unsorted JSON keys.
- Including extra whitespace in the serialized JSON string.
- Serializing time in UTC rather than in the declared TAI-compatible integer field.
- Renaming fields such as `M_kg` or `T_TAI_unix_s`.
- Changing the object metadata included in the serialized structure.

Each of these conditions can produce a different digest even when the conceptual Earth state appears unchanged. For this reason, conformance testing should always include the canonical JSON string itself rather than validating only the final hash [cite:94][cite:96].

Suggested implementation note

A manuscript or technical specification that includes this annex should state explicitly that the Earth reference test vector is normative for QHD-1.0 validation. This gives reviewers and implementers a

This model and documentation are provided for non-commercial use only under the CC BY-NC-SA 4.0 license.

No commercial use, profiling, or exploitation of IP logs or trade secrets is permitted.

See full “License and Conditions of Use” notice for details.

In case of conflict or ambiguity, the most restrictive interpretation in favor of non-commercial use, user privacy, and protection of trade secrets shall prevail.

Author: Carlos Daniel Borrego Romero

stable benchmark for determining whether an implementation is genuinely compliant with the declared protocol [cite:55][cite:56][cite:57].

Annex status

This annex may be used as Annex B in a scientific manuscript, technical report, or institutional proof-of-concept package. Together with the Earth validation record in Annex A, it provides both the empirical reference case and the implementation-facing validation logic needed for independent replication of the QHD base case [cite:55][cite:56][cite:57][cite:96].

LICENSE AND CONDITIONS OF USE.

The content is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) license:

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

By using this content and model, you agree to:

Provide appropriate credit, include a link to the above license, and indicate if changes were made.

Not use the material or any derivative works for commercial purposes.

Distribute any adaptations or derivative works only under the same CC BY-NC-SA 4.0 license.

Not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Data logging and trade secrets policy.

The design, implementation, and deployment of this model, as well as any derivatives, do not authorize the collection, storage, or exploitation of IP addresses, unique device identifiers, or other technical traceability data for commercial profiling, targeted advertising, or any other economic exploitation.

Any system implementing this model must limit the logging and retention of IP addresses and other identifiable data strictly to what is necessary to:

(a) maintain technical security and service integrity; and

(b) comply with applicable legal obligations (for example, any minimum log retention required by law).

It is prohibited to use this model or its derivatives to:

capture, store, or exploit trade secrets, proprietary know-how, or confidential information of users or third parties for any purpose other than providing the technical service described in this documentation; or train or improve closed, commercial systems using data that contains confidential information of third parties without their prior, explicit, written consent.

Public institutions, space agencies, research organizations, and universities may use this model and documentation for non-commercial research, validation, mission analysis, training, and internal operational support, provided that such use complies with the CC BY-NC-SA 4.0 license and does not convert QHD-based work products into closed, proprietary commercial offerings.

---+EOD+---